

# Perspectivism in federated practice

Martijn Houtman and Pieter Wisse

For pedagogical reasons, that is, in order to concentrate on some principles, we'll start reasoning here from an undifferentiated view of reality.<sup>1</sup>

In recognition of our limits at ... recognition, what only stands out is limited to a horizon as our ... limit. In a model drawn up with Metapattern,<sup>2</sup> this horizon is represented by a thick, horizontal line at the top.

## 1. Divergence (inverse inheritance)

A most general application of differentiation entails assuming that reality is made up of things, say, objects. At type-level, figure 1 models such a merely objectified reality.

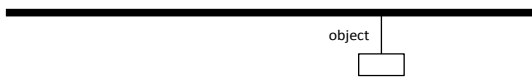


figure 1: only seeing objects in reality.

A next step in differentiation could be to classify each instance of object. That would require distinguishing types, enabling subsequent typing; see figure 2.

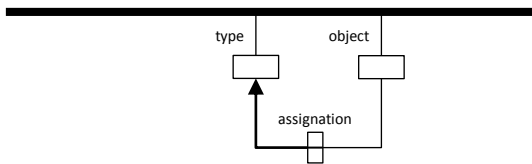


figure 2: assigning type(s) to objects.

Please note that one and the same object instance may be assigned different type instances. This really sets off perspectivism. For we can regard a particular object as, for example, a bicycle, or real estate, et cetera.

What makes such appearances different are their distinct(ive) behaviors as – assumed being – displayed by their particular properties. Relevant property types are then taken to be determined by (object) type. Through (object) type, that is, for each (object) assignment a template is available, allowing for adding one or more values for each relevant property. Such further differentiation is sketched in figure 3.

If strictly conceptual differentiation is at stake, only serving a theoretical purpose, the model of figure 3 suggests what is both necessary and sufficient.

---

<sup>1</sup> This paper attempts to conceptualize some applications of KnitbITs in support of federated information exchange. KnitbITs is the infrastructural platform for information exchange on offer by Information Dynamics (Netherlands). As such, KnitbITs allows for implementation of information resources & services as modeled with Metapattern (also developed by Information Dynamics).

<sup>2</sup> For access to English-language documentation, see [Metapattern, handbook infrastructure for information exchange](#) (Information Dynamics, with classified quotations referring to original articles, papers et cetera).

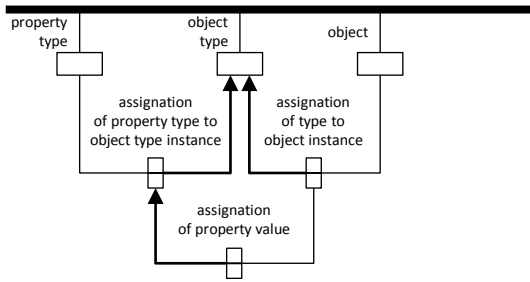


figure 3: extending perspectivism to include explicit typical variety of object properties.

However, what we model must also be practically feasible. So, we have to reckon with distributed information. At the time of design, even especially so, we have to have – from acknowledging to establishing – a federation of information sets (also read: registers) in mind.

It follows that an instance of object type may constitute yet another horizon, one that is – in spatial terms – closer (1). Then, from such a more singularized perspective, the assignment of type to an instance of object, say, reappears within the more limited sphere as – still – generally conceived object instance (2). In figure 4 an attempt is made to visually demonstrate the shift.

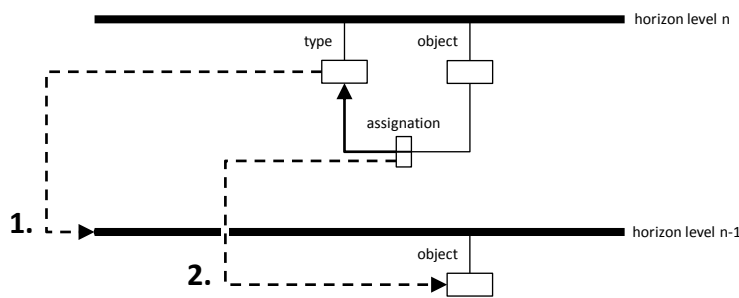


figure 4: connecting levels in a hierarchy of perspectives.

It turns out, not surprisingly, that such differentiation through horizons comes down to our earlier work labeled information roundabout.<sup>3</sup>

What should be clear is that for an object, (only) its continuous identity is assumed.

We may, and invariably will, interpret an object from different perspectives (also read: motives). Yet, when we also believe – in a pragmatic sense – that different, necessarily perspectival, interpretations pertain to the same object,<sup>4</sup> what those perspectives structurally have in common is an overall identity for ‘their’ object.

In addition, please note that shifting horizons annex levels for typing is open-ended in both directions, exemplifying Metapattern’s characteristic decompositions directed at specifying context, respectively behavior.<sup>5</sup>

<sup>3</sup> In Dutch, we wrote *Stelselmatig overzicht via informatiesleutels* (2013). It contains references to earlier papers on information roundabout, also in Dutch.

<sup>4</sup> According to Metapattern, a particular perspective ‘situates’ the object, limiting what we can interpret at a time to a situated object with its relevant situated behavior.

<sup>5</sup> Across horizons, in this way Metapattern may in fact also be used to implement so-called type inheritance according to object orientation. Within a single horizon, however, Metapattern even rules out such inheritance;

## 2. Intermezzo on metasystematics

Now, through such verticality, actual object instances occur within each horizon. How its properties read from different perspectives, may then be reported by following up, both within a single horizon and across horizons, on the object's so-called continuous identity.<sup>6</sup> Let's call this the direct, or immediate, method for relating perspectives.

Then, an indirect, or intermediate, method applies when relationships have been made explicit only for types. How types 'form' systems could be modeled separately from models oriented at object instances. For the left-hand side of figure 5, we copied a system model "for necessary and sufficient specification."<sup>7</sup>

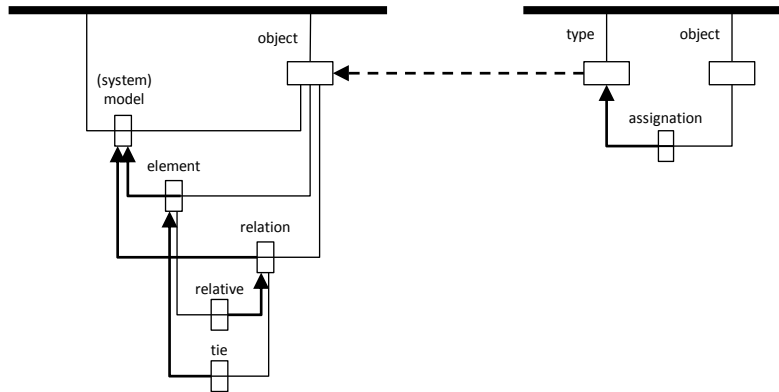


figure 5: juxtaposing type-level relationships.

In fact, the right-hand side of figure 5 need not be limited to a single horizon. Suggesting an open-ended scope for instances, the model on the right should be replaced with – the model in – figure 4, above.

The intermediate method cannot exist independently,<sup>8</sup> but may be applied to complement what we've labeled the immediate method.

What we've treated so far, are divergent – structures for – behaviors; what starts out as 'being' equal (object), also requires differentiation (assignment).

---

in a well-differentiated model, a signature's contextualizations yield disjunct intexts corresponding, as the semiotic ennead suggests, with disjunct behaviors of an object's plural situationalizations.

<sup>6</sup> It is called nil identity in *Metapattern; context and time in information models* (Addison-Wesley, 2001, author: Pieter Wisse). Explanation relevant to his concept is reproduced in *The pattern of metapattern: ontological formalization of context and time for open interconnection* (in: *PrimaVera*, working paper 2004-01, Amsterdam University, 2004).

<sup>7</sup> See figure 6 in *An intermediary metasystematics* (2013) by Pieter Wisse and Jan van Til with the authors aiming at demonstrating "the system approach to interdependency." That way, the reach of the intermediate method for relating properties may be extended indefinitely, too.

How that may be accomplished in practice, might in turn benefit from how handling perspectivism is proposed here. Rather than including "objectifications" (see figure 7 in *An intermediary metasystematics*) within a single horizon, a hierarchy of horizons could be established for the purpose, allowing for types according to different horizons (figure 5, above, right-hand side) referring to objects for systematization (left-hand side). Of course, such possibilities for intermediate derivation et cetera extend beyond, multiple or not, object-oriented inheritance with super- and subclasses, only.

<sup>8</sup> Currently still popular approaches to artificial intelligence, however, seem to exclusively start from, to put it as a pleonasm, type-level abstractions. Especially when limited to a single perspective, they can only remain sterile. As singularity of perspective is often implicit, it is all the more difficult to overcome. Its proponents resist the necessary paradigm shift all the more stubbornly because awareness is lacking of paradigms being at stake.

### 3. Convergence (inverse polymorphism)

We are calling convergent when more or less the opposite occurs, i.e. starting from differently typed objects as in figure 6. It could be, however, that instances of both types at least share – some of their structure for – behavior. As a matter of separation of concerns, in this case avoiding redundancy, such common behavior should be uniquely positioned, too. For this purpose, within the original horizon a classification of behavior types is added.<sup>9</sup> Assigning a typed object to a behavior type yields a node (assignment) to carry a reference, possibly to another horizon et cetera. It follows that different nodes within one horizon may all refer to the same horizon that was also added to supply the structurally more detailed behavior being separated as a concern.

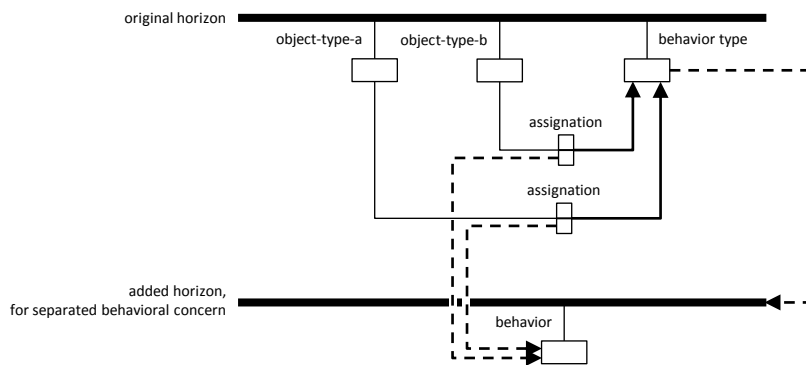


figure 6: converging subsequent behaviors.

Please note the, say, functional equivalence between the models as presented in figures 4 and 6. When modeling from scratch, figure 4 exemplifies the preferred approach. Anyway, it keeps a conceptual model optimally ... conceptual. Often, though, there are established practices to consider. It usually turns out that change can only be effected stepwise. Under such circumstances, the approach as suggested with figure 6 indicates local modifications for gradually improving cohesion.

### 4. Dynamically balancing approaches

Information exchange in the so-called networked society can be facilitated realistically only with a federated infrastructure. Such an infrastructure will continue to develop, necessarily exhibiting a changing mixture of di- and convergent approaches to perspectivism.

dr Martijn Houtman (Information Dynamics) developed KnitBITs for implementing context-oriented information systems modeled with Metapattern.

dr ir Pieter Wisse (Information Dynamics) developed Metapattern, a method and language for context-oriented conceptual modeling.

September 2014 © Information Dynamics, Voorburg (Netherlands)

<sup>9</sup> Several assignments are possible for one and the same instance of a particular typed object. In fact, that is the whole point of this exercise. What is still lacking, and not to be confused with instance-level references, is what determines selection between those different assignments and subsequent behaviors. In Metapattern's semiotics, this requires explicit contexts. Figure 6 is therefore already conceptually incomplete, but as it stands should highlight what we mean with convergence (as opposed to what we've demonstrated, above, as divergence).

## Some additional guidelines

Models are normally documented one horizon at a time. It might be useful to suggest that other levels are involved to constitute a federation of information sets/registers; see figure 7 for some minimalistic symbols (in this context).

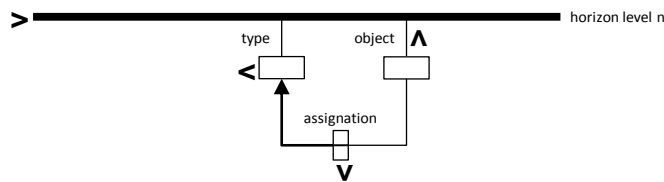


figure 7: signs for indicating hierarchy (when applicable).

For conceptual modeling, we recommend first of all capturing variety with(in) a single model (horizon). The model in figure 3 is ... typical for this approach. Next, the modeling choice concerns whether or not to commit to merging object type with object.

Suppose only two types are relevant, a and b respectively. Instead of subsequent parameterization of type, as in figure 2 and known as postcoordination, an initial separation could be established. Now, a template is an immediate part of each already typical object. The simplicity of figure 8, as compared with figure 3, might be deceptive, though.

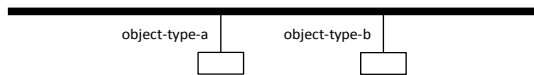


figure 8: precoordination of object types.

Its advantage lies in directness, familiarity. When only a (very) small number of object types need consideration, and their number doesn't change, why not?

But as the scope of – infrastructural – information resources increases, such assumptions do not hold. Then should come the – by now rhetorical – question of whether or not resources are distributed. For any serious purposes, they are.

As modelers/designers, we need to realize that abstract and concrete have become treacherous ideas. For it all depends on scope/scale. Why not call it perspective?

What has to operate at a large scale (broader perspective), may seem abstract when viewed from a much smaller scale (narrower perspective). But, in fact, everything must be exhaustively concrete at its relevant scale (from its relevant perspective), period. Otherwise, it simply doesn't ... work.

With opportunities for infrastructure regardless of scale, Metapattern's principle of recursive contextual differentiation facilitates disambiguating variety. Temporal differentiation is included, because variety is also dynamic.